

# DBMS ASSIGNMENT 4

## 1. order by clause:

Query: select \* from manager order by salary;

Output:

The screenshot shows a database management tool interface. On the left, there's a 'SCHEMAS' panel with a tree view containing 'rdb', 'restaurant db', 'sakila', 'sys', and 'world'. The main area displays a query: `1 • select * from manager order by salary;`. Below the query, there's a 'Result Grid' showing the output of the query. The grid has columns: 'manager id', 'name', 'gender', 'phone number', 'salary', and 'annualsalary'. The data is sorted by salary in ascending order. Below the grid, there's an 'Output' section showing the execution details: 'Action Output', 'Time: 17:42:28', 'Action: select \* from manager order by salary LIMIT 0, 1000', 'Message: 5 row(s) returned', and 'Duration / Fetch: 0.000 sec / 0.000 sec'.

manager id	name	gender	phone number	salary	annualsalary
101	sruthi	female	933774742	30000	360000
102	mandeep	male	725252434	30000	360000
100	ram	male	927737372	40000	480000
103	krishna	male	937365532	50000	600000
104	sandhya	female	726353353	50000	600000

## 2. Group by and having:

Query: select \* from manager group by having salary > 30000;

Output:

The screenshot shows the same database management tool interface. The query in the main area is: `1 • select * from manager group by gender,salary having salary > 30000;`. The 'Result Grid' shows the output of this query, which is filtered to show only rows where the salary is greater than 30000. The grid has the same columns as the first screenshot. The 'Output' section shows the execution details: 'Action Output', 'Time: 17:44:07', 'Action: select \* from manager group by gender,salary having salary > 30000 LIMIT 0, 1000', 'Message: 3 row(s) returned', and 'Duration / Fetch: 0.000 sec / 0.000 sec'.

manager id	name	gender	phone number	salary	annualsalary
100	ram	male	927737372	40000	480000
103	krishna	male	937365532	50000	600000
104	sandhya	female	726353353	50000	600000

### 3. Aggregate functions:

#### a. using count:

Query: `select count(*) from manager where salary > 30000;`

Output:

The screenshot shows the SQL Developer interface. The query editor contains the query: `select count(*) from manager where salary > 30000;`. The result grid displays a single row with the value 6. The action output pane shows the execution details: `select count(*) from manager where salary > 30000 LIMIT 0, 1000` returned 1 row(s) in 0.093 seconds.

count(*)
6

#	Time	Action	Message	Duration / Fetch
1	19:18:16	select count(*) from manager where salary > 30000 LIMIT 0, 1000	1 row(s) returned	0.093 sec / 0.000 sec

#### b. using average:

Query: `select avg(salary) from manager where gender = 'male';`

Output:

The screenshot shows the SQL Developer interface. The query editor contains the query: `select avg(salary) from manager where gender = 'male';`. The result grid displays a single row with the value 52500.0000. The action output pane shows the execution details: `select avg(salary) from manager where gender = 'male' LIMIT 0, 1000` returned 1 row(s) in 0.109 seconds.

avg(salary)
52500.0000

#	Time	Action	Message	Duration / Fetch
1	19:22:51	select avg(salary) from manager where gender = 'male' LIMIT 0, 1000	1 row(s) returned	0.109 sec / 0.000 sec

### c. using sum:

Query: select sum(salary) from manager;

Output:

The screenshot shows a database client interface with a 'SCHEMAS' panel on the left listing 'rdb', 'restaurant db', 'sakila', 'sys', and 'world'. The main query editor contains the SQL statement: `select sum(salary) from manager;`. Below the editor, the 'Result Grid' displays a single row with the value 395000. The 'Output' panel at the bottom shows the execution details for 'Result 3'.

#	Time	Action	Message	Duration / Fetch
1	19:29:27	select sum(salary) from manager LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec

### d. using max:

Query: select max(salary) from manager;

Output:

The screenshot shows the same database client interface as above, but with the SQL query changed to: `select max(salary) from manager;`. The 'Result Grid' now displays a single row with the value 70000. The 'Output' panel at the bottom shows the execution details for 'Result 4'.

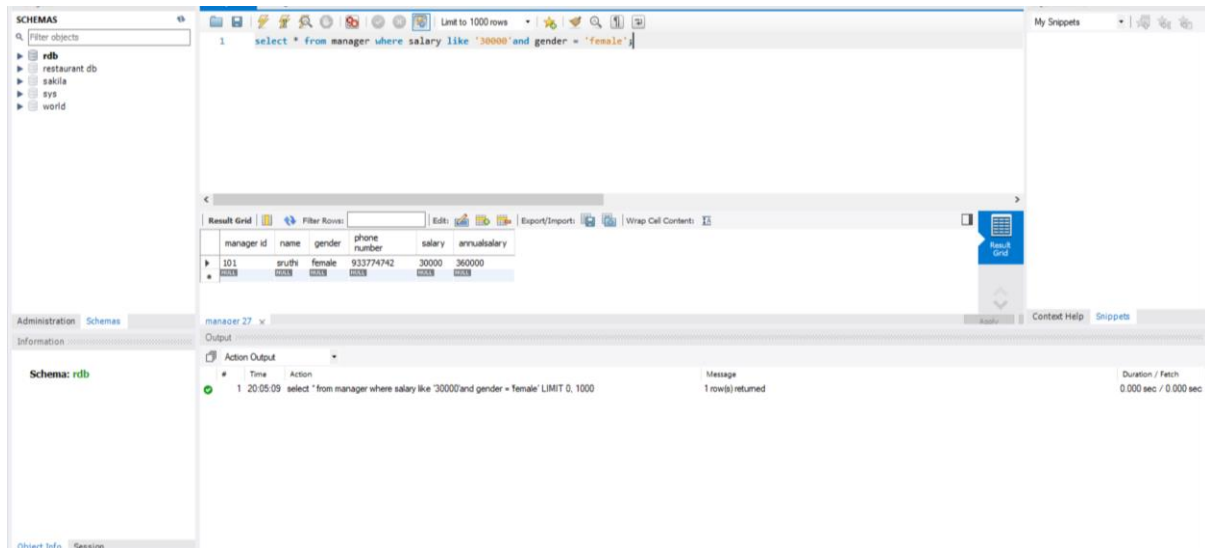
#	Time	Action	Message	Duration / Fetch
1	19:31:53	select max(salary) from manager LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

## 4. Logical operators especially with like:

### a. using LIKE and AND operator:

Query: select \* from manager where salary like '30000' and gender = 'female';

Output:



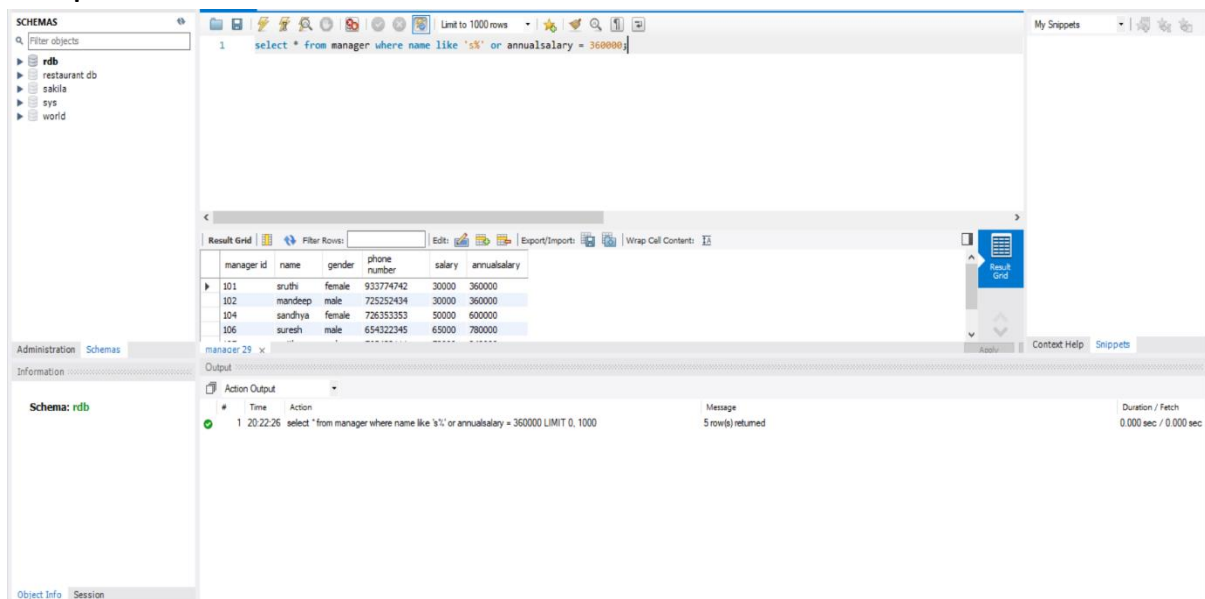
The screenshot shows the SQL Developer interface. The query editor contains the query: `select * from manager where salary like '30000' and gender = 'female';`. The result grid displays one row of data. The action output pane shows the query execution details, including the time (20:05:09) and the message "1 row(s) returned".

manager id	name	gender	phone number	salary	annualsalary
101	sruthi	female	933774742	30000	360000

### b. using LIKE and OR operator:

Query: select \* from manager where name like 's%' or annualsalary = 360000;

Output:



The screenshot shows the SQL Developer interface. The query editor contains the query: `select * from manager where name like 's%' or annualsalary = 360000;`. The result grid displays five rows of data. The action output pane shows the query execution details, including the time (20:22:26) and the message "5 row(s) returned".

manager id	name	gender	phone number	salary	annualsalary
101	sruthi	female	933774742	30000	360000
102	mandeep	male	725252434	30000	360000
104	sandhya	female	726353353	50000	600000
106	suresh	male	654322345	65000	780000
107	...	...	...	...	...

## 5. Nested Queries:

### a. simple subquery:

Query:

```
select name from customer where customer_id = (select customer_id from customer where city = 'hyderabad');
```

Output:

The screenshot displays a database management interface. On the left, a 'SCHEMAS' pane shows a tree view with 'rdb' expanded, containing 'restaurant db', 'sakila', 'sys', and 'world'. The main editor shows a SQL query: `1 select name from customer where customer_id = (select customer_id from customer where city = 'hyderabad');`. Below the query, a 'Result Grid' shows a single row with the name 'hemant'. The bottom panel, titled 'customer 1 x', shows the 'Table: customer' with columns: 'customer\_id' (int, AI PK), 'name' (varchar(45)), 'gender' (varchar(45)), 'phone number' (int), 'city' (varchar(45)), and 'bill\_id' (int). The 'Action Output' pane shows the query execution details: '1 01:28:07 select name from customer where customer\_id = (select customer\_id from customer where city = 'hyderabad') ... 1 row(s) returned' with a duration of '0.000 sec / 0.000 sec'.

### b. simple subquery:

Query:

```
select customer.name, customer.city from customer where customer.customer_id IN (select customer_id from customer where customer_id = '201') and customer.bill_id = '46';
```

## Output:

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane lists databases: rdb, restaurant db, sakila, sys, and world. The main query editor contains the following SQL query:

```
1 select customer.name, customer.city from customer where customer.customer_id IN (select customer_id from customer where customer_id = '201')
```

The 'Result Grid' shows the following data:

name	city
hemant	hyderabad

Below the query editor, the 'customer 1' table is selected. The 'Output' pane shows the query execution details:

#	Time	Action	Message	Duration / Fetch
1	01:43:39	select customer.name, customer.city from customer where customer.customer_id IN (select customer_id from ...	1 row(s) returned	0.000 sec / 0.000 sec

## c. Multiple subquery:

### Query:

select customer\_id, name, city from customer where bill\_id > (select avg(bill\_id) from customer where gender = (select gender from customer where customer\_id = '206' and '205'));

## Output:

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane lists databases: rdb, restaurant db, sakila, sys, and world. The main query editor contains the following SQL query:

```
1 select customer_id, name, city from customer where bill_id > (select avg(bill_id) from customer where gender = (select gender from customer where customer_id = '206' and '205'));
```

The 'Result Grid' shows the following data:

customer_id	name	city
204	prem	kochi
205	vijay	punjab
206	sruthi	indore

Below the query editor, the 'customer 2' table is selected. The 'Output' pane shows the query execution details:

#	Time	Action	Message	Duration / Fetch
1	02:17:58	select customer_id, name, city from customer where bill_id > (select avg(bill_id) from customer where gender ...	3 row(s) returned	0.000 sec / 0.000 sec

#### d. Multiple subquery:

Query:

```
select name, bill_id, gender from customer where customer_id =  
(select customer_id from customer where city = 'mumbai');
```

Output:

The screenshot shows a database management tool interface. On the left, there is a 'SCHEMAS' panel with a tree view showing databases like 'rdb', 'restaurant\_db', 'sakila', 'sys', and 'world'. Below this is a 'Table: manager' section showing its columns: 'manager\_id' (int, AI PK), 'name' (varchar(45)), 'gender' (varchar(20)), 'phone number' (int), 'salary' (int), and 'annualsalary' (double). The main area displays a SQL query: `select name, bill_id, gender from customer where customer_id = (select customer_id from customer where city = 'mumbai');`. Below the query, a 'Result Grid' shows one row of data: 'vamshi', '45', and 'male'. At the bottom, an 'Output' section shows the execution details: '1 03:38:10 select name, bill\_id, gender from customer where customer\_id = (select customer\_id from customer where city ... 1 row(s) returned' with a duration of '0.000 sec / 0.000 sec'. The status bar at the bottom indicates 'Query Completed'.

# END

**(Combined Group 21 and 23)**

p.shiva hemanth 18bcs066

B.praveen 18bcs020

s.Guna shekar 18bcs090

p.chandhan 18bcs063

